

Music Ontology Tutorial 1

This tutorial shows how a simple RDF document describing a music event can be written using the Music Ontology. It introduces some basic concepts of RDF using the RDF syntax called 'Turtle'.

<http://www.dajobe.org/2004/01/turtle/>

Turtle documents should be saved with the extension .n3 or .ttl .

For the purposes of this tutorial, I wish to publish information about a (fictitious) music event I once attended: Wayne Shorter (saxophone), Herbie Hancock (piano) and Jaco Pastorius (bass guitar), playing together at the 'Shunt' venue near London Bridge Station; I recall that one number they played in the gig was 'Summertime'.

This involves three basic steps:

- a) writing a description of the event as an RDF document;
- b) validating the document;
- c) publishing the document.

a) Write a description of the event

Using a text editor, declare some namespaces that we use throughout this document.

<http://www.w3.org/TR/REC-xml-names/>

1. Declare a prefix for the Music Ontology:

```
@prefix mo:<http://purl.org/ontology/mo/>.
```

[Note that complete statements in Turtle end with a full stop “.”.]

2. Declare a prefix for the whole document:

```
@prefix:<#>.
```

[For example, with a document at http://example.org/my_document.n3, “:thing” would be interpreted as: http://example.org/my_document.n3#thing.]

3. Declare an identifier for our performance using the Music Ontology's concept of “Performance”:

```
:perf a mo:Performance.
```

4. Associate a known musical genre with the performance:

```
:perf mo:genre <http://dbpedia/resource/Jazz>.
```

5. Create sub-events p1, p2, p3 for the event (one for each performer) using the standard OWL event ontology:

```
@prefix event: <http://purl.org/NET/c4dm/event.owl#>.
:perf event:sub_event :p1, :p2, :p3.
```

[Note that to declare multiple items which in this case are all 'sub-events', we simply separate them by commas.]

6. Now we associate a description with each sub-event:

```
:p1 mo:performer <http://dbpedia.org/resource/Wayne\_Shorter>; mo:instrument
<http://dbpedia.org/resource/Saxophone>.
:p2 mo:performer
<http://dbtune.org/musicbrainz/resource/artist/27613b78-1b9d-4ec3-9db5-fa0743465fdd>; mo:instrument <http://purl.org/ontology/mo/mit#Grand\_Piano>.
:p3 mo:performer <http://dbpedia.org/resource/Jaco\_Pastorius>;
mo:instrument :i3
```

[The description has two parts: performer and instrument, separated by semicolon. The performer URIs are those obtained using the semantic web search engines <http://sindice.com/> (p1 and p3) and <http://dbtune.org/musicbrainz/> (p2). This allows us to obtain more information about the performers without the burden of writing it ourselves. Similarly, it is useful to fall back on existing information on the instruments, as in the case of p1 and p2; for p3 we define a special instrument type 'i3' – see below.]

7. Define a special instrument type i3:

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>.
:i3 a mo:Instrument; rdfs:label "weird bass guitar".
```

[Note that we defined another prefix “rdfs”, which refers to “rdf schema”, which defines commonly used properties such as “label”.]

8. Associate a place and time with the performance.

```
:perf event:place <http://dbpedia.org/resource/London_Bridge_station>.

@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
:perf dc:date "2008-06-08"^^xsd:date.
```

[See “datatype” at <http://www.dajobe.org/2004/01/turtle/>. Note that the caret signs (^^) tell the parser how to interpret the string containing the date.]

9. Associate a listener with the performance, in our case this is Yves Raimond.

```
:perf mo:listener <http://data.semanticweb.org/person/yves-raimond>.
```

10. Lastly, we mention the song “Summertime” being played as part of the performance:

```
:perf event:sub_event :s1.
@prefix tl: <http://purl.org/NET/c4dm/timeline.owl#>.
:s1 mo:performed_work :summertime.
:summertime a mo:MusicalWork; rdfs:label "Summertime"; mo:time
[tl:onTimeLine <http://matthiasmauch.de/_musik/Summertime.mma.rdf#tl>].
```

[We need to define a new sub-event s1, this time representing a song. We need the timeline ontology (see <http://motools.sourceforge.net/timeline/timeline.html>) because the description of the song we link to relies on a temporal sequence concept.]

b) Validate the document

Use the online parser at <http://librdf.org/parse> to check that the RDF syntax is correct. Alternatively download and/or compile one of the many tools available, e.g. Raptor (<http://librdf.org/raptor/>).

c) Publish the document

Simply upload it to a public webspace that can be accessed by web crawlers. Note that your server needs to serve the proper MIME type because it will not be derived from the file extension alone (see <http://www.dajobe.org/2004/01/turtle/>).